Σχόλια και παρατηρήσεις για το lab 5

Μεταγλώττιση: Ο κώδικας θα πρέπει να μεταγλωττίζεται χωρίς **warnings**. Χαρακτηρισμός “μη ικανοποιητικό” σημαίνει ότι είχατε τουλάχιστον ένα warning. Κώδικας που παράγει λάθη (errors) κατά τη μεταγλώττιση βαθμολογείται με FAIL.

Σχόλια: Πρέπει να υπάρχουν γενικά σχόλια προγράμματος στην αρχή (ονοματεπώνυμο, ημερομηνία, σύντομη περιγραφή). Σχόλια μέσα στον κώδικα πρέπει να βάζετε όταν δίνουν μια πληροφορία που δεν είναι προφανής. Για παράδειγμα, ένα σχόλιο που λέει /* ektypwsi apotelesmatos */ είναι κακό γιατί βλέπουμε την printf. Αν έχετε σχόλια που εξηγούν τα ονόματα των μεταβλητών τότε μάλλον έχετε άσχημα ονόματα μεταβλητών… Αλλά ένα σχόλιο δίπλα, για παράδειγμα, στις διάφορες ποσότητες της δεύτερης άσκησης που να λέει σε τι μονάδα είναι αυτά τα νούμερα είναι χρήσιμο γιατί δίνει μια επιπλέον πληροφορία που δε φαίνεται από τον κώδικα.

"Μέτριο" σημαίνει ότι τα σχόλια είναι ελλιπή ή ανακριβή ή υπερβολικά και μη χρήσιμα.

"Μη ικανοποιητικό" σημαίνει ότι δεν υπάρχουν σχόλια.

Στοίχιση: Ομοια με τις απαιτήσεις των **lab3, lab4**. Προσοχή στη χρήση κενών γραμμών για διαχωρισμό "ενοτήτων" στο πρόγραμμα (αλλά όχι πάνω από μία διαδοχικά), στη χρήση χαρακτήρα space πριν και μετά από τελεστές (να μην είναι όλα κολλημένα σε κάθε έκφραση) και να μην εκτείνεται κώδικας ή σχόλια πέρα τις 80ής στήλης. Ο κώδικας πρέπει να είναι στοιχισμένος σωστά, με ιδιαίτερη προσοχή στη στοίχιση εμφωλευμένων for/while/do while/if/switch κλπ.

Χαρακτηρισμός “μέτρια” γενικά σημαίνει ότι έχετε κάποια στοίχιση αλλά δεν είναι ακριβώς σωστή, για παράδειγμα έχετε περισσότερα από ένα tab, ή αντί για tab έχετε 1 ή 2 κενά.

Χαρακτηρισμός “μη ικανοποιητική” σημαίνει ότι δεν έχετε στοιχίσει τον κώδικά σας ή η στοίχιση είναι πολύ ασυνεπής (άλλες γραμμές οκ, άλλες όχι), ειδικά όσον αφορά τις δομές ελέγχου/επανάληψης.

Διαβάστε ξανά το φυλλάδιο “Αρχές καλού προγραμματισμού” και δείτε την στοίχιση στις λύσεις του εργαστηρίου που θα αναρτηθούν στο e-class.

Ονόματα και τύποι Μεταβλητών: Οπως πάντα, θέλουμε περιγραφικές μεταβλητές με κατάλληλους τύπους. Δώστε ιδιαίτερη προσοχή στις μεταβλητές που χρησιμοποιούμε ως μετρητές σε for loop: όταν είναι δυνατό, προτιμάμε να τις επαναχρησιμοποιούμε παρά να ορίζουμε καινούργιες.

Χρησιμοποιείτε double για αριθμούς κινητής υποδιαστολής ή όπου δεν προσδιορίζεται επακριβώς για να έχουν μεγαλύτερη ακρίβεια οι υπολογισμοί. Τα ονόματα θα πρέπει να είναι σύντομα αλλά περιγραφικά. Τα ονόματα const μεταβλητών θα πρέπει να είναι γραμμένα με κεφαλαία.

Δώστε ιδιαίτερη προσοχή στις μεταβλητές που χρησιμοποιούμε ως μετρητές σε for/while loop: όταν είναι δυνατό, προτιμάμε να τις επαναχρησιμοποιούμε παρά να ορίζουμε καινούργιες. Οι μετρητές είναι οι μόνες μεταβλητές που μπορούν να ονομαστούν με ένα μόνο γράμμα (π.χ. i, j, k)

"Μέτρια" ονόματα είναι dist, hgt ή ονόματα const που είναι γραμμένα με μικρά.

"Μη ικανοποιητικά" είναι d, g.

Επαναλήψεις/if: Προσέξτε τις συνθήκες. Ειδικά για for, πρέπει να προσέχετε ιδιαίτερα τα όρια ώστε να γίνεται ο κατάλληλος αριθμός επαναλήψεων. Αλλο το for (i=0; i<size; i++) κι άλλο το for (i=0; i<=size; i++). Επίσης, συνίσταται να χρησιμοποιείτε πάντα { } ακόμη κι αν το σώμα μιας εντολής for αποτελείται από μία μόνο εντολή. Στην άσκηση 1 της Πέμπτης είναι δεκτό είτε switch είτε if. Τέλος, όταν γνωρίζετε εν των προτέρων το πλήθος επαναλήψεων (πχ. πλήθος αυτοκινήτων, φοιτητών, αστερίσκων), είναι προτιμότερο να χρησιμοποιείτε for και όχι while.

Εξοδος προγράμματος: Η έξοδος του προγράμματος πρέπει να είναι ΑΚΡΙΒΩΣ ίδια με την εκφώνηση.